

Course: WebDev – Django Start.

Homework - 4



Theme: Introduction To Django

Level: beginner

Instructor: Mikhail Nakonechnyi

Due Date: 6th March, 2020

Problem 1. Создание сайта. Этап 1 - Подготовка

На начальном этапе необходимо провести некую подготовительную работу. Такая работа предполагает сбор информации о целевой аудитории и целях ресурса. Иначе говоря, вы должны придумать **ИДЕЮ**. Наш первый сайт будет блогом. Основываясь на результатах сбора информации, вы должны сделать вывод о том, какую тематику для своего сайта выбрать: новости (спорт, мода, наука...), тематический блог (кулинария, автомобили, шитье и тд). И следующим моментом будет придумать название, которое подчеркивает тематику сайта.

Примечание: дополнительные ресурсы для помощи:

- [Как узнать интересы целевой аудитории](#)
- [Как собрать информацию об аудитории Методы сбора информации и инструменты анализа](#)
- [О чем вести блог](#)
- [Темы для блога](#)
- [Еще темы](#)
- [И еще темы](#)

Problem 2. Создание сайта. Этап 2 - Контент

Основываясь на своей тематике сайта, написать/сочинить как минимум 1 статью. Название статьи должно быть не более 200 символов. Статья должна содержать не менее 3000 символов (с пробелами). Для статьи можно найти несколько тематических картинок широкого формата. Для работы воспользуйтесь **Google Docs**. Для того, чтобы вести учет своих работ - создайте папку на гугл диске с названием сайта, а в ней папку **Статьи**. Самый простой вариант хранить статьи подставляя номер статьи перед ее названием.

Контент – основа вашего сайта. Будь это авторские статьи или пользовательские статьи – пользователь в первую очередь приходит за информацией, и мы должны в приятной форме её преподнести. На этом этапе необходимо иметь представление, какие разделы будут на вашем сайте. Например если будет страница «О сайте» – что на ней найдет посетитель? Рекомендую проверять на орфографические ошибки (хотя бы при помощи MS Word). Остальные подробности о написании статей узнаете на занятии.

Примечание: ссылку на гугл документ отправить мне в телеграме. Я не приму работу скопированную под чистую из интернета.

Problem 4. Команды Bash

Некоторые команды для работы с командной строкой терминала в UNIX/Linux или терминалом powershell(Windows), которые необходимо знать и использовать все время для работы:

- **ls** - отображение файлов и папок находящихся в директории
 - **ls -l** - отображение файлов и папок с дополнительными параметрами
 - **ls -la** - отображение файлов и папок с дополнительными параметрами и скрытыми файлами
 - **cd PATH** - перейти в каталог по заданному пути
 - **cd ..** - перейти в "родительский" каталог
 - **cd ~** - перейти в "домашнюю" директорию
 - **cd -** - вернуться на одно действие назад (местонахождение)
 - **cd path/to/documents/** - пример "пути"
 - **clear** - очистить окно терминала
 - **mkdir FOLDER_NAME** - создать папку с названием
 - **mv path/to/file.txt newfile.html** - переместить файл / папку или переименовать файл / папку
 - **cp path/to/file.txt newfile.html** - скопировать файл или папку
 - **rm path/to/file.txt** - удалить файл по адресу
 - **rm -rf directory/** - удалить папку
 - **touch index.html** - создать файл index.html
-

Problem 4. Установка Python

Примечание: инструкции и описания ниже относятся к терминалу Linux (в частности, к нашему IDE для работы). Django написан на Python. Нам нужен Python, чтобы сделать что-нибудь в Django. Начнем с его установки! Мы хотим самую свежую версию Python 3. Для установки нужно ввести команду:

```
1 $ apt-get install python3
```

Если уже установлена версия 3.4 или более высокая, она должна подойти. Для проверки можно воспользоваться следующей командой:

```
1 $ python3 --version
```

Problem 5. Настройка виртуального окружения

Перед установкой Django мы должны установить крайне полезный инструмент, который поможет содержать среду разработки в чистоте. Можно пропустить этот шаг, но лучше этого не делать. Использование лучших рекомендаций с самого начала убережёт от многих проблем в будущем! Итак, создадим виртуальное окружение (оно также называется `virtualenv`). **Virtualenv** будет изолировать настройки Python/Django для каждого отдельного проекта. Это значит, что изменения одного сайта не затронут другие сайты, которые будут разрабатываться. Всё, что нужно сделать – найти директорию, в которой мы создадим `virtualenv`; домашний каталог вполне подойдёт. Мы будем использовать директорию нашего проекта в домашнем ка-

талоге:

```
1 ~/ $ cd django
```

Мы создадим виртуальное окружение под именем **myenv**.
В общем случае команда будет выглядеть так:

```
1 ~/django $ python3 -m venv myenv
```

Указанная выше команда создаст директорию *myenv* (или другую, в зависимости от выбранного имени), которая будет содержать виртуальное окружение (по сути – набор файлов и папок). Будет видно, что *virtualenv* запущено, когда появится префикс (*myenv*) в начале командной строки. Для запуска виртуального окружения, необходимо выполнить следующую команду в папке *django*:

```
1 ~/django $ source myenv/bin/activate
```

При работе с виртуальным окружением команда *python* будет автоматически обращаться к правильной версии языка, так что можно использовать просто *python* вместо *python3*.
Теперь мы будем хранить все важные зависимости в одном месте.

Problem 5. Знакомство с Django

Перейдем к установке Django!
